OBJECT ORIENTED PROGRAMMING CONCEPTS

LAB MANUAL

1. Given that an EMPLOYEE class contains following members:
Data members : EMPLOYEE_number, EMPLOYEE_name, Basic,DA,IT and Net_sal
Member functions : To read the data, to calculate Net_sal and to print data members.

Write a C++ program to read the data of N EMPLOYEEs and compute Net_sal of each EMPLOYEE. (DA=52% of Basic and income tax(IT)=30% of grass salary).

```
#include<iostream.h>
#include<conio.h>
#define size 25

class EMPLOYEE     //CLASS DEFINITION
{
                char emp_num[10],
                emp_name[25]; float basic, da ,
                it, net_sal;
        public:
                void
                getdata();
                void netsal();
                void display();
};

void EMPLOYEE :: getdata() //READING THE DATA
{ cout<<"\nenter EMPLOYEE
        number:"; cin>>emp_num;
        cout<<"enter EMPLOYEE
        name:"; cin>>emp_name;
        cout<<"enter basic salary:";
        cin>>basic;
}

void EMPLOYEE :: netsal()   //CALCULATING THE NET SALARY
{ da=(.52*basic); float
        gsal=basic+da;
        it=(.30*gsal);
        net_sal=gsal-it;
}
void EMPLOYEE::display()//DISPLAYING THE OUTPUT
```

```
{ cout<<"\n\nEMPLOYEE
        number:"<<emp_num
        <<"\nEMPLOYEE name:"<<emp_name
            <<"\nnetsalary:"<<net_sal<<endl;
}

void main()          //MAIN FUNCTION
{
        clrscr();
        EMPLOYEE
        obj[size]; int n;

        cout<<"\nenter number of
        EMPLOYEEs:"; cin>>n;

        for(int i=0;i<n;i++)
        {
                obj[i].getdata();
                obj[i].netsal();
        }

        for(i=0;i<n;i++)
          obj[i].display();
    getch();
}
```

OUTPUT:

```
        enter number of
        EMPLOYEEs:2 enter
        EMPLOYEE number:15
        enter EMPLOYEE
        name:mandaara enter basic
        salary:2000

        enter EMPLOYEE
        number:25 enter
        EMPLOYEE name:madhura
        enter basic salary:3000

        EMPLOYEE
        number:15
        EMPLOYEE
        name:mandaara
        netsalary:2128
```

EMPLOYEE number:25
EMPLOYEE name:madhura
netsalary:3192

2. Define a STUDENT class with USN, Name and Marks in 3 tests of a subject. Declare an array of 10 STUDENT objects. Using appropriate functions, find the average of two better marks for each STUDENT. Print USN, Name and average marks of all the STUDENTs.

```
#include<iostream.
h>
#include<conio.h>
#define size 25

class STUDENT
{ int     m1,m2,m3;
          charusn[20],name[20];
  public:
           void getdata();
           void putdata();
};

void STUDENT :: getdata()
{ cout<<"\n usn:"; cin>>usn;
          cout<<"\n name:";
          cin>>name; cout<<"\n
          enter three marks:";
          cin>>m1>>m2>>m3;
}

void STUDENT :: putdata()
{ int min; float
          avg;
          min=0;
           if((m1<=m2) && (m1<=m3))
                    min=m1;
            else
                  if((m2<=m1) && (m2<=m3))
                      min = m2;
             else
                      min = m3;
           avg=(float)((m1+m2+m3)-min)/2.0; cout<<"usn-
           >"<<usn<<endl
               <<"name->"<<name<<endl
```

```
                <<"avg marks-
  >"<<avg<<endl<<endl; }
void main()
{
        STUDENT
         obj[size]; int i,n;
         clrscr();
         cout<<"enter the no of STUDENTs:";
         cin>>n;
         for(i=1;i<=n;i++)
         { cout<<"\n enter"<<”STUDENT ” <<i<<”
                  data:”<<endl; obj[i].getdata();
          }
         cout<<"STUDENT information:"<<endl;
         for(i=1;i<=n;i++)
                obj[i].putdata();
         getch()
; }
```

Output:

```
        enter the number of
        STUDENTs: 2 enter STUDENT
        1 data:

                                usn:1db02cs30
                                name:madhura enter
                                three marks:12 13
                                12

        enterSTUDENT    2    data:    usn:1db02is25
                                name:mandaara
                                enter three marks:24
                                21 24

        STUDENT      information:      usn      -
                                >1db02cs30
                                name          ->
                                madhura   avg
                                marks -> 12.5

                                usn           ->
                                1db02is25
                                name          ->
```

mandaara avg
marks -> 24

03. Write a C++ program to create a class called as COMPLEX, & implement the following by overloading the function ADD which returns the COMPLEX number.

i) ADD(a,s2) - Where a is an integer(Real Part) & s2 is a complex number.
ii) ADD(s1,s2) - Where s1 and s2 are complex numbers.iii) Display the result by overloading the operator <<.

```
#include<iostream.h>
#include<conio.h>

class COMPLEX
{ float real, imag;
public:
        void getdata(float , float );

/* Function overloading: Functions are differ by Number and
        Type of arguments*/ friend COMPLEX add(int ,
        COMPLEX ); friend COMPLEX add(COMPLEX ,COMPLEX);

/* Overloding the operator Insertor(<<), and Extractor( >>) operator
  overloading functions should be Friends to that class */ friend
  ostream& operator <<(ostream &,COMPLEX &);
};

void COMPLEX :: getdata(float n1, float n2)
{ real=n1;
         imag=n
         2;
}

ostream& operator <<(ostream& print, COMPLEX& comp)
{ print<<comp.real<<" +i
       "<<comp.imag;
          return print;
}

COMPLEX add(int a, COMPLEX s2)
{
  COMPLEX result;
  result.real=num+s2.real;
  result.imag=s2.imag;
```

```
   return(result);
}
COMPLEX add( COMPLEX s1, COMPLEX s2)
{
   COMPLEX result;
   result.real=s1.real+s2.real;
   result.imag=s1.imag+s2.imag;
   return(result);
}

void main()
{ int n1,n2,n3;
          COMPLEX c1,c2,c3,c4;
          clrscr();

          cout<<"\nEnter the real and imaginary part of the Ist
          complex number :"; cin>>n1>>n2; c1.getdata(n1,n2);

          cout<<"\nEnter the real and imaginary part of the IInd
          complex number :"; cin>>n1>>n2; c2.getdata(n1,n2);

          cout<<"\nEnter the integer to be added to the Ist
          complex number :"; cin>>n3; c3=add(c1,c2);
          c4=add(n3,c2);

          cout<<"\n\nSum of two complex numbers "<<endl
             <<"\t c1 : "<<c1<<endl<<"\t c2 : "<<c2<<endl
             <<endl<<"\t IS : "<<c3<<endl
             <<"\nSum of integer with complex number is"<<endl
            <<"\tc2: "<<c2<<endl<<"\tinteger: "<<n3<<endl
             <<"\tIS   : "<<c4 <<endl;
          getch();
}
RUN 1
------

Enter the real and imaginary part of the Ist complex number : 1 2
Enter the real and imaginary part of the IInd complex number: 3 4
Enter the integer to be added to the Ist complex number        : 5


Sum of two complex numbers
```

c1 : 1 +i
2 c2 : 3
+i 4

IS : 4 +i 6

Sum of integer with complex number is

c2 : 3 +i 4
integer: 5
IS : 8 +i 4

RUN 2.------
Enter the real and imaginary part of the Ist complex number :5 0Enter the real and imaginary part of the IInd complex number :0 8
Enter the integer to be added to the Ist complex number5 Sum of two complex numbersc1 : 5 +i 0 c2 : 0 +i 8
IS : 5 +i 8

Sum of integer with complex number isc2 : 0 +i 8

integer: 5
IS : 5 +i
8

04. Write a C++ program to create a class called LIST (linked list) with member functions to insert an element at the front as well as to delete an element from the front of the list. Demonstrate all the functions after creating a list object.

```
#include<iostream.h>
#include<conio.h>

struct list
{ int info; struct
        list *link;
}; typedef struct list

NODE;

class linklist
{
        NODE *first;
```

```
public:          //Constructor - for initializing each NODE to zero.
                 linklist(void)
                  { first=NULL;
                   } void
                 insertnode(int);
                 void delnode();
                 void display();
};
                                          // Function to insert the new NODE
void linklist :: insertnode(int ele)
{
  NODE *newnode = new NODE;
  newnode->info = ele;
  newnode->link = first;
  first=newnode;
}
                                      // Function to delete the NODE
void linklist :: delnode()
{
  clrscr();
  NODE *temp=first;
  if(first==NULL) cout<<"THE
  LIST IS EMPTY!!!!!"; else
     { cout<<"THE ELEMENT DELETED IS:-
        "<<first->info; first=first->link;
       delete temp; cout<<"\n\
       tDELETION SUCCESSFUL!!!!";
     }
   getch(
  ); }

void linklist :: display()
{
  clrscr();
  NODE *temp=first;
  if(first==NULL)
     cout<<"THE LIST IS EMPTY !!!!!! ";
  else
  { cout<<"THE ELEMENTS IN THE LINKEDLIST
      ARE:-";
      while(temp!=NULL)
```

```
     { cout<<temp-
       >info<<"      ";
       temp=temp->link;
      }
   }
  getch()
; }

void main()
{               int
    ele,choice;
    linklist obj;
    clrscr();
   do
   {
         clrscr();
         cout<<"MENU:-\n"; cout<<"\t(1)INSERTION \n\
         t(2)DELETION \n\t(3)DISPLAY\n\t(4)EXIT";
         cout<<"\nENTER YOUR CHOICE :";
         cin>>choice;

         switch(choice)
         {
           case 1:       clrscr(); cout<<"ENTER THE ELEMENT TO
                         BE INSERTED :"; cin>>ele;
                         obj.insertnode(ele); break;
            case 2:       obj.delno
            case 3:       de();
            case 4:       break;


                          obj.displa
                          y();
                          break;
                        cout<<"THANK YOU!!!!";
                         break;
        }
     } while(choice!
   =4);
   getch();
}
```

```
  MENU:(1)INSERTION (2)DELETION (3)DISPLAY (4)EXIT

ENTER YOUR CHOICE : 1
```

ENTER THE ELEMENT TO BE INSERTED : 2

ENTER YOUR CHOICE : 1 ENTER THE ELEMENT TO BE INSERTED : 20

ENTER YOUR CHOICE : 1 ENTER THE ELEMENT TO BE INSERTED : 40

ENTER YOUR CHOICE : 3 THE ELEMENTS IN THE LINKEDLIST ARE:- 40 20 2

ENTER YOUR CHOICE :2
THE ELEMENT DELETED IS:- 40

DELETION SUCCESSFUL!!!!

ENTER YOUR CHOICE :2
THE ELEMENT DELETED IS:- 20 DELETION SUCCESSFUL!!!!

ENTER YOUR CHOICE :2 THE ELEMENT DELETED IS:- 2
DELETION SUCCESSFUL!!!!

ENTER YOUR CHOICE :2
THE LIST IS EMPTY!!!!!

ENTER YOUR CHOICE :3
THE LIST IS EMPTY !!!!!!
ENTER YOUR CHOICE :4 THANK YOU!!!!

5. Write a C++ program to create a template function for quick sort and demonstrate the sorting of integers, doubles and character datatypes.

```
#include<iostream.h>
#include<conio.h>

                    // Function template for exchanging(any type)
template <class
t> void swap(t
&x, t &y)
{ t
  temp;
  temp=
  x;
  x=y;
  y=tem
  p;
}

                    // Function template for sorting(any type)
template <class t1>
void quick(t1 a[], int low, int high)
```

```
{ t1
  key;
  int i,j,flag=0;
  if(low<high)
  { key=a[low];
          i=low+1;
          j=high;
          while(!flag)
          { while((a[i] <= key ) &&
                  (i<high)) i++;
                  while(a[j]>key)
                          j--;
                  if(i<j)
                       swap(a[i],a[j]
                       );
                  else
                        flag=
                        1;
            }
          swap(a[low],a[j])
          ;
          quick(a,low,high
          -1);
          quick(a,low+1,h
          igh);
     }
    return;
  }
void main()
{ int i,j,num,a[30],choice;
         double b[30];

        cout<<"MENU:-\n";
         cout<<"\t1. INTEGERS\n\t2. DOUBLES\n\t3. Exit\n";

         clrscr();
         cout<<"ENTER YOUR CHOICE ...:";
         cin>>choice;
          switch(choice)
         { case 1: cout<<"Enter how many elements are there in the
           integer array=>"; cin>>num; cout<<"Enter the Integer
           numbers  : ";
                   for(i=0;i<num;i++)
                        cin>>a[i];
```

```
                    quick(a,0,num-1) ;
                    cout<<"\n The Sorted List
                    is... : "; for(i=0;i<num;i+
                    +) cout<<a[i]<<"\t";
                    break;

          case 2: cout<<"Enter how many elements are there in double
                  vector array=> "; cin>>num; cout<<"Enter the
                  Floating point numbers : ";
                  for(i=0;i<num;i++)
                      cin>>b[i];
                  quick(b,0,num-1)            ;
                  cout<<"\nThe  Sorted  List
                  is...  :  ";  for(i=0;i<num;i+
                  +) cout<<b[i]<<"\t";
                  break;

          default : cout<<"THANK YOU!!!!";
                    break;
     }
   getch();
}
```

Run 1.

MENU:1.
INTEGERS
2.DOUBLES
3.Exit

ENTER YOUR CHOICE ... : 1
Enter how many elements are there in the

integer array=>5 Enter the Integer numbers :

-3 5 -25 0 2 The Sorted List is...

: -25 -3 0 2 5

Run 2.

MENU:1.
INTEGERS
2.DOUBLES
3.Exit

ENTER YOUR CHOICE ... : 2
Enter how many elements are therein double vector
array=> 4 Enter the Floating point numbers : 2.5 -
3.6 -1.345 1.25

The Sorted List is... : -3.6-1.345 1.25 2.5

06. Write a C++ program to create a class called STACK using an array of integers. Implement the following operations by overloading the operators + & -.

i) s1 = s1 + element - Where s1 is a object of the class STACK &element is a integer to be pushed on top of the stack.
ii) s1 = s1-- Where s1 is a object of the class STACK, -- operator pops the element from the stack.

Handle the STACK empty and STACK full conditions. Also display the status and contents of the stack after each operation, by overloading the operator <<.

```
#include<iostream.h>
#include<conio.h>

class STACK
{ int a[100], size, top;
 public :
                    //Constructor with one argument to intialize the
                  stack to -1 //And the Max size of the stack to n that
                                given by the user.
        STACK(int n)
        { top=-1;
                size=
                n;
        }

        void display();
        friend STACK operator +
        (STACK , int); friend STACK
        operator --(STACK ); friend int
        overflow(STACK); friend int
        empty( STACK );
        friend ostream& operator <<(ostream& ,STACK );
};

STACK operator +(STACK s1, int ele)
{
```

```
   clrscr(); s1.a[+
   +s1.top]=ele;
   return (s1);
}

STACK operator --(STACK s1)
{
   clrscr();
   cout<<" \nTHE ELEMENT "<<s1.a[s1.top--]
   <<" HAS BEEN POPPED OUT OF THE
   STACK."; getch(); return(s1);
}

         //Functions to check the overflow &
underflow int overflow( STACK s1)
{ if(s1.top==(s1.siz
   e-1) ) return(1);
  else
return(0); }

int empty( STACK s1)
{ if(s1.top==-1)
  return(1);
 else
return(0); }

ostream& operator <<(ostream& print,STACK s1)
{ if(empty(s
 1))
  {
    clrscr();
    print<<"\n\tTHE STACK IS
   EMPTY!!!!!"; getch(); }
  else
  {
   clrscr();
   print<<"\nSTACK
   STATUS       :-";       for(int
   i=s1.top;             i>=0;i--)
   print<<"\t"<<s1.a[i];
   getch();
   }
 return(prin
t); }
```

```
void main()
{ int choice,element,n;
        clrscr();
         cout<<"\nEnter the size of the stack : ";
         cin>>n;
         STACK
         s1(n); do
         {
                clrscr();
                cout<<"MENU :-\n(1)PUSH\n(2)POP\n(3)DISPLAY\
                n(4)EXIT"; cout<<"\nENTER YOUR CHOICE: ";

                cin>>choice;
                switch(choice)
                { case 1:cout<<"\nENTER THE ELEMENT TO BE
                 PUSHED :";
                        int element;
                        cin>>element;
                        if(overflow(s1))
                        {
                                clrscr();
                                cout<<"\n\t STACK OVERFLOW!!!!! ";
                                getch();
                        }
                        else
                                 s1=s1+elem
                                 ent;
                       cout<<s1;
                       break;

                case 2: if(empty(s1))
                       {
                                clrscr();
                                cout<<"\n THE STACK IS EMPTY!!!!";
                                getch();
                       }
                       else s1=s1--;
                       cout<<s1;
                       break;

                case 3: cout<<s1;
                       break;

                case 4: cout<<"\nTHANKYOU!!!!!";
```

```
                        break;
                }
        } while(choice!
    =4);
  getch();
}
```

Output:

```
Enter the size of the stack :      3

MENU :-

        (1)PUSH
        (2)POP
        (3)DISPLAY
        (4)EXIT

ENTER YOUR CHOICE: 1
ENTER THE ELEMENT TO BE PUSHED :2
STACK STATUS :- 2

ENTER YOUR CHOICE: 1
ENTER THE ELEMENT TO BE PUSHED :4
STACK STATUS :- 4       2

ENTER YOUR CHOICE: 1
ENTER THE ELEMENT TO BE PUSHED :6
STACK STATUS :- 6     4     2

ENTER YOUR CHOICE: 1
                STACK OVERFLOW!!!!! ;
STACK STATUS :- 6     4     2

ENTER YOUR CHOICE: 2
THE ELEMENT 6 HAS BEEN POPPED OUT OF THE STACK.
STACK STATUS :- 4     2

ENTER YOUR CHOICE: 2
THE ELEMENT 4 HAS BEEN POPPED OUT OF THE STACK.
STACK STATUS :- 2

ENTER YOUR CHOICE: 2
THE ELEMENT 2 HAS BEEN POPPED OUT OF THE
                STACK. THE STACK IS EMPTY!!!!!
```

```
ENTER YOUR CHOICE: 2
                THE STACK IS EMPTY!!!!1
STACK STATUS :- THE STACK IS
EMPTY!!!!!

ENTER YOUR CHOICE: 4
                 THANKYOU!!!!!
```

07. Write a C++ program, to create a class called as DATE. Accept two valid dates in the form of dd/mm/yyyy. Implement the following by overloading the operators - & +. Display the results by overloading the operator << after each operation.

i) no_of_days = d1 - d2 where d1 & d2 are Date objects. Assume d1>= d2 andno_of_days is an integer.

ii) d1 = d1 + no_of_days where d1 is a DATE object and no-of-days is an integer.

```
#include <iostream.h>
#include <conio.h>

class date
 {
            int dd,mm,yy;

       public:
            void read()
              { cin>>dd>>mm>>
                    yy;
              } int operator -
            (date); date
            operator + (int);
            friend ostream &operator<<(ostream &put,date &d);
 };
 int
 b[13]={0,31,29,31,30,31,30,31,31,30,31,3
 0,31}; int
 a[13]={0,31,28,31,30,31,30,31,31,30,31,3
 0,31};

 int date::operator-(date d2)
       { date
         res;
         int nod,noly=0,temp=d2.yy;

         for
         (temp;temp<yy;temp
```

```
        ++) if(temp%4==0)
        noly++;

        temp=yy; for
        (temp;temp>d2.yy;tem
        p--)
        if(temp%4==0)
              noly++;

        res.dd=dd-d2.dd;
        if(res.dd<0)
        { res.dd=dd+a[m
              m];
              res.mm=mm--;
        } res.mm=mm-
        d2.mm;
        if(res.mm<0)
        { res.mm=mm+1
        2; res.yy=yy--; }
        res.yy=yy-d2.yy;
        if(yy<0)
            return -1;
        nod=res.dd+(res.yy*365);

        int months=d2.mm;
        for(int
        i=1;i<=res.mm;i++)
              nod+=a[months++];

        nod+=noly;
    return
   nod; }

date date::operator+(int ndays)
       { date
        d;
        d.dd=dd; d.mm=mm; d.yy=yy;
        for(int i=1;i<=ndays;i++)
              {
               d.dd++; if(d.yy
               %4==0)
                { if(d.dd>b[d.m
                     m])
                       { d.dd=1;
```

```
                        d.mm++;
                       }
                  }
                 else
                 { if(d.dd>a[d.m
                    m])
                       { d.dd=1;
                         d.mm++;
                       }
                 }
                 if(d.mm>1
                 2)
                       { d.mm=1;
                       d.yy++;
                       }
               }
         return d;
   }
ostream &operator<<(ostream &print,date &d)
  {

        print<<d.dd<<"/"<<d.mm<<"/"<<d.yy;
        return
  print; }

void main()
  { date d1,d2;
        int ndays;
        clrscr();
        cout<<"Enter two dates, date1 should be greater
        than date2\n\n"; cout<<"Enter Date1 (dd mm
        yyyy)::"; d1.read(); cout<<"Enter Date2 (dd mm
        yyyy)::"; d2.read();
        clrscr();
        ndays=d1-d2;
        cout<<"\n\nNumber of Days between the\n\n\t\tDate1 ::
        “<<d1<<"\nand"
            <<"\t\tDate2 :: " <<d2<<" is =>"<<ndays;
        cout<<"\n\n\nEnter no. of days to be added to
        the Date1 ::"; cin>>ndays; d1=d1+ndays;
        cout<<"\n\n\t\tDate1 after adding "<<ndays<<" days
        is ::"<<d1;

        getch();
```

```
 }
```

RUN 1.

ENTER THE FIRST DATE IN THE DD/MM/YYYY FORMAT : 1 1 2000 ENTER THE SECOND DATE IN THE SAME FORMAT : 1 1 1999

THE NO. OF DAYS IN BETWEEN THE TWO YEARS ARE : 365 ENTER THE NO. OF DAYS TO BE ADDED TO THE FIRST DATE : 0

THE DATE AFTER 0 DAYS IS : 1 / 1 / 2000

RUN 2.
-----ENTER THE FIRST DATE IN THE DD/MM/YYYY FORMAT : 1 1 1997 ENTER THE SECOND DATE IN THE SAME FORMAT : 1 1 1996

THE NO. OF DAYS IN BETWEEN THE TWO YEARS ARE : 366 ENTER THE NO. OF DAYS TO BE ADDED TO THE FIRST DATE : 5

THE DATE AFTER 5 DAYS IS : 6 / 1 / 1997

08. Write a C++ program, to create a class called MATRIX using two dimensional arrays of integers. Implement the following by overloading the operators = which checks the compatibility of two matrices to be added and subtracted. Perform the following by overloading + and - operators. Display the result by overloading the operator <<.

```
    if ( m1 = m2 )
     {
        m3 = m1 + m2 ;
        m4 = m1 - m2 ;
     }

    else
            Display 'error'.
```

Where m1,m2,m3,m4 are MATRIX objects.

```
#include<iostream.h>
#include<conio.h>

class MATRIX
{ int row,column,a[10]
        [10];

public :
        MATRIX(int r, int c)
        { row=r;
        column=
        c; }
```

```
        void getmatrix(); int
        operator ==(MATRIX);
        MATRIX operator +
        (MATRIX); MATRIX
        operator -(MATRIX);
        friend ostream& operator <<(ostream&, MATRIX &);
};

void MATRIX :: getmatrix()
{ for(int i=0;i<row;i+
   +) for(int
  j=0;j<column;j++)
    { cout<<"
    [ "<<i<<","<<j<<" ] :";
    cin>>a[i][j];
    }
  }
int MATRIX :: operator ==(MATRIX m1)
{ if((m1.row == row) && (m1.column
  == column)) return(1);
 else
return(0); }


MATRIX MATRIX :: operator +(MATRIX m1)
{
        MATRIX m2(row,column);
         for(int i=0 ; i<row ; i++)
          for(int j=0 ; j<column ; j++)
                  m2.a[i][j]=a[i][j]+m1.a[i][j];
          return m2;
}

MATRIX MATRIX :: operator -(MATRIX m1)
{
         MATRIX m2(row,column);
         for(int i=0 ; i<row ; i++)
          for(int j=0 ; j<column ; j++)
                  m2.a[i][j]=a[i][j]-m1.a[i][j];
         return
m2; }


ostream& operator <<( ostream& print, MATRIX &res)
```

```
{ for(int i=0;i<res.row;i+
        +)
         { for(int j=0;j<res.column;j+
                  +)
                   print<<res.a[i][j]<<"";
                   print<<endl ;
          }
        return
        print;
}
void
main() {

 int m,n,p,q;
 clrscr();

 cout<<"Enter the order of the First
 matrix => "; cin>>m>>n; MATRIX
 m1(m,n);

 cout<<"Enter the order of the Second matrix=>
 "; cin>>p>>q;
 MATRIX m2(p,q);

 if(m1 == m2)
 { cout<<"\nEnter the elements of First matrix
         :- \n"; m1.getmatrix();

         cout<<"\nEnter the elements of Second matrix :-\n";
         m2.getmatrix();

         MATRIX m3(m,n), m4(m,n);

         m3=m1+m2; m4=m1-m2; cout<<m1<<"
   +      \n"<<m2<<"      =      \n"<<m3<<endl;
   cout<<m1<<" - \n"<<m2<<" = \n"<<m4; }

 else
  cout<<"\nTHE TWO MATRICES ARE NONCOMPATIBLE FOR
  ADDITION OR
               SUBTRACTION!!!!";
 getch();
}
Run 1.
------
```

```
Enter the order of the First matrix => 2 1
Enter the order of the Second matrix => 2 1

THE TWO MATRICES ARE NONCOMPATIBLE FOR ADDITION OR
SUBTRACTION!!!!


Run 2.
------
Enter the order of the First matrix => 2 2
Enter the order of the Second matrix=> 2 2

Enter the elements of First
 matrix :[ 0,0 ] : 2
 [ 0,1 ] : 3
 [ 1,0 ] : 4
 [ 1,1 ] : 5

Enter the elements of Second
 matrix :[ 0,0 ] : 6
 [ 0,1 ] : 5
 [ 1,0 ] : 4
 [ 1,1 ] : 8

2   3
4   5
   +
6   5
4   8
   =
8   8
 8 13


2   3
4   5
   -
6   5
4   8
   =
-4 -2
 0 -3
```

09. Write a C++ program, to create a class called as OCTAL which has the characteristics of an octal number. Implement the following by

writing an appropriate constructor and an overloaded operator +. Display the OCTAL object by overloading the operator <<. Also display the values of k & y.

i) OCTAL h = x; Where h is OCTAL object and x is an integer. ii) int y = h + k; here h is an OCTAl and k is an integer.

```
#include<iostream.h>
#include<conio.h
>
#include<math.h
> class OCTAL
{ int oct[15] , count;
  public:
          OCTAL(int);
          int operator +(int);
          friend ostream& operator <<(ostream& ,OCTAL)
};

OCTAL :: OCTAL(int x)
{ int
 i=0,rem,a[15]
 ; while(x!=0)
 { rem=x%8;
 x=x/8; a[i+
 +]=rem; }
 count=i; int
 n=count-1;
 for(i=0;i<count
 ;i++)
 {
  oct[i]=a[n];
  n--;
 }
}

int OCTAL :: operator +(int k)
{ int x=0;int
  j=count-1; for(int
  i=0;i<count;i++)
  { x=x+oct[j]*pow(8,
        i);
        j--;
```

```
   }
   return(x+
   k);
}
ostream& operator <<(ostream& print,OCTAL o)
{ for(int
 i=0;i<o.count;i++)
 print<<o.oct[i];
 return
print; }
void
main()

{
 clrscr();
 int x,k,y=0; cout<<"\nEnter the Integer value
 of x (In Decimal) : "; cin>>x;
 OCTAL h=OCTAL(x); cout<<"The corresponding Octal
 Equivalent of      "<< x <<"   is       : "<<h; cout<<"\n\
 nEnter the integer to be added to the Octal value : ";
 cin>>k; y=h+k; cout<<"\t\t\t"<<h<<" (Octal) + "<<k<<"
 (Decimal) = "<<y<<" (Decimal)"; getch();
}
```

Run 1.

```
------
Enter the Integer value of x (In Decimal)    : 10
The corresponding Octal Equivalent of 10  is       : 12

Enter the integer to be added to the Octal value : 10
                        12 (Octal) + 10 (Decimal) = 20
(Decimal) Run 2.
------
Enter the Integer value of x (In Decimal)    : 8
The corresponding Octal Equivalent of 8    is       : 10

Enter the integer to be added to the Octal value : 2
                        10 (Octal) + 2 (Decimal) = 10 (Decimal)
```

Run 3.

```
------
Enter the Integer value of x (In Decimal)    : 5
The corresponding Octal Equivalent of 5    is       : 5
```

Enter the integer to be added to the Octal value : 6

5 (Octal) + 6 (Decimal)= 11 (Decimal)

10. Write a C++ program, to create a class QUEUE, with add and delete member functions. Using it, implement a QUEUE of integers and doubles. Demonstrate the implementation by displaying the status and the content of the QUEUE after every operation.

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>

template <class
t> class QUEUE
{ t a[100]; int
        rear;
 public:
        QUEUE()
        { rear
        =0; }

        void add(t);
        void   del();
        void
        display();
};

template <class t> void
QUEUE <t> :: add( t ele)
{ a[+
  +rear]=ele
  ;
}

template <class t>
void QUEUE <t> ::
del()
{ if(rear=
 =0)
 { cout<<"QUEUE IS EMPTY !!!!!\n\n PRESS ANY KEY TO
 CONTINUE....."; getch(); }

 else
 { t ele=a[1]; for(int
       i=1;i<=rear;i++)
       a[i]=a[i+1];
```

```
        rear--;
……
```

1. Given that an EMPLOYEE class contains following members:
Data members : EMPLOYEE_number, EMPLOYEE_name, Basic,DA,IT and Net_sal
Member functions : To read the data, to calculate Net_sal and to print data members.

Write a C++ program to read the data of N EMPLOYEEs and compute Net_sal of each EMPLOYEE. (DA=52% of Basic and income tax(IT)=30% of grass salary).

```
#include<iostream.h>
#include<conio.h>
#define size 25

class EMPLOYEE     //CLASS DEFINITION
{
                char emp_num[10],
                emp_name[25]; float basic, da ,
                it, net_sal;
        public:
                void
                getdata();
                void netsal();
                void display();
};

void EMPLOYEE :: getdata() //READING THE DATA
{ cout<<"\nenter EMPLOYEE
        number:"; cin>>emp_num;
        cout<<"enter EMPLOYEE
        name:"; cin>>emp_name;
        cout<<"enter basic salary:";
        cin>>basic;
}

void EMPLOYEE :: netsal()   //CALCULATING THE NET SALARY
{ da=(.52*basic); float
        gsal=basic+da;
        it=(.30*gsal);
        net_sal=gsal-it;
}
void EMPLOYEE::display()//DISPLAYING THE OUTPUT
```

```
{ cout<<"\n\nEMPLOYEE
        number:"<<emp_num
        <<"\nEMPLOYEE name:"<<emp_name
            <<"\nnetsalary:"<<net_sal<<endl;
}

void main()          //MAIN FUNCTION
{
        clrscr();
        EMPLOYEE
        obj[size]; int n;

        cout<<"\nenter number of
        EMPLOYEEs:"; cin>>n;

        for(int i=0;i<n;i++)
        {
                obj[i].getdata();
                obj[i].netsal();
        }

        for(i=0;i<n;i++)
          obj[i].display();
    getch();
}
```

OUTPUT:

```
        enter number of
        EMPLOYEEs:2 enter
        EMPLOYEE number:15
        enter EMPLOYEE
        name:mandaara enter basic
        salary:2000

        enter EMPLOYEE
        number:25 enter
        EMPLOYEE name:madhura
        enter basic salary:3000

        EMPLOYEE
        number:15
        EMPLOYEE
        name:mandaara
        netsalary:2128
```

EMPLOYEE number:25
EMPLOYEE name:madhura
netsalary:3192

2. Define a STUDENT class with USN, Name and Marks in 3 tests of a subject. Declare an array of 10 STUDENT objects. Using appropriate functions, find the average of two better marks for each STUDENT. Print USN, Name and average marks of all the STUDENTs.

```
#include<iostream.
h>
#include<conio.h>
#define size 25

class STUDENT
{ int     m1,m2,m3;
          charusn[20],name[20];
  public:
           void getdata();
           void putdata();
};

void STUDENT :: getdata()
{ cout<<"\n usn:"; cin>>usn;
          cout<<"\n name:";
          cin>>name; cout<<"\n
          enter three marks:";
          cin>>m1>>m2>>m3;
}

void STUDENT :: putdata()
{ int min; float
          avg;
          min=0;
           if((m1<=m2) && (m1<=m3))
                    min=m1;
            else
                  if((m2<=m1) && (m2<=m3))
                      min = m2;
             else
                      min = m3;
           avg=(float)((m1+m2+m3)-min)/2.0; cout<<"usn-
           >"<<usn<<endl
               <<"name->"<<name<<endl
```

```
            <<"avg marks->"<<avg<<endl<<endl;
  } void
main() {
        STUDENT
         obj[size]; int i,n;
         clrscr();
         cout<<"enter the no of STUDENTs:";
         cin>>n;
         for(i=1;i<=n;i++)
         { cout<<"\n enter"<<"STUDENT " <<i<<"
                  data:"<<endl; obj[i].getdata();
          }
          cout<<"STUDENT information:"<<endl;
          for(i=1;i<=n;i++)
                obj[i].putdata();
          getch()
; }
```

Output:

```
        enter the number of
        STUDENTs: 2 enter STUDENT
        1 data:

                                  usn:1db02cs30
                                  name:madhura enter
                                  three marks:12 13
                                  12

        enterSTUDENT    2    data:    usn:1db02is25
                                  name:mandaara
                                  enter three marks:24
                                  21 24

        STUDENT     information:     usn     -
                                  >1db02cs30
                                  name         ->
                                  madhura   avg
                                  marks -> 12.5

                                  usn          ->
                                  1db02is25
                                  name         ->
                                  mandaara  avg
                                  marks -> 24
```

03. Write a C++ program to create a class called as COMPLEX, & implement the following by overloading the function ADD which returns the COMPLEX number.

i) ADD(a,s2) - Where a is an integer(Real Part) & s2 is a complex number.

ii) ADD(s1,s2) - Where s1 and s2 are complex numbers.iii) Display the result by overloading the operator <<.

```
#include<iostream.h>
#include<conio.h>

class COMPLEX
{ float real, imag;
public:
        void getdata(float , float );

/* Function overloading: Functions are differ by Number and
        Type of arguments*/ friend COMPLEX add(int ,
        COMPLEX ); friend COMPLEX add(COMPLEX ,COMPLEX);

/* Overloding the operator Insertor(<<), and Extractor( >>) operator
  overloading functions should be Friends to that class */ friend
  ostream& operator <<(ostream &,COMPLEX &);
};

void COMPLEX :: getdata(float n1, float n2)
{ real=n1;
         imag=n
         2;
}

ostream& operator <<(ostream& print, COMPLEX& comp)
{ print<<comp.real<<" +i
       "<<comp.imag;
          return print;
}

COMPLEX add(int a, COMPLEX s2)
{
   COMPLEX result;
   result.real=num+s2.real;
   result.imag=s2.imag;
   return(result);
```

```
}
COMPLEX add( COMPLEX s1, COMPLEX s2)
{
   COMPLEX result;
   result.real=s1.real+s2.real;
   result.imag=s1.imag+s2.imag;
   return(result);
}

void main()
{ int n1,n2,n3;
          COMPLEX c1,c2,c3,c4;
          clrscr();

          cout<<"\nEnter the real and imaginary part of the Ist
          complex number :"; cin>>n1>>n2; c1.getdata(n1,n2);

          cout<<"\nEnter the real and imaginary part of the IInd
          complex number :"; cin>>n1>>n2; c2.getdata(n1,n2);

          cout<<"\nEnter the integer to be added to the Ist
          complex number :"; cin>>n3; c3=add(c1,c2);
          c4=add(n3,c2);

          cout<<"\n\nSum of two complex numbers "<<endl
             <<"\t c1 : "<<c1<<endl<<"\t c2 : "<<c2<<endl
             <<endl<<"\t IS : "<<c3<<endl
             <<"\nSum of integer with complex number is"<<endl
            <<"\tc2: "<<c2<<endl<<"\tinteger: "<<n3<<endl
            <<"\tIS: "<<c4 <<endl;
         getch()
; }
```

RUN 1

```
Enter the real and imaginary part of the Ist complex number : 1 2
Enter the real and imaginary part of the IInd complex number: 3 4
Enter the integer to be added to the Ist complex number        : 5
Sum of two complex numbers

        c1 : 1 +i
        2 c2 : 3
        +i 4

        IS : 4 +i 6

Sum of integer with complex number is

        c2 : 3 +i 4
        integer: 5
        IS  : 8 +i 4
```

RUN 2.

```
------
Enter the real and imaginary part of the Ist complex number :5 0
Enter the real and imaginary part of the IInd complex number :0 8
Enter the integer to be added to the Ist complex
number5 Sum of two complex numbers

        c1 : 5 +i
        0 c2 : 0
        +i 8

        IS : 5 +i 8

Sum of integer with complex number is

        c2  : 0 +i 8
        integer: 5
        IS  : 5 +i 8
```

04. Write a C++ program to create a class called LIST (linked list) with member functions to insert an element at the front as well as to delete an element from the front of the list. Demonstrate all the functions after creating a list object.

```
#include<iostream.h>
#include<conio.h>

struct list
{ int info;
        struct list *link;
}; typedef struct list

NODE;

class linklist
{
        NODE *first;

public:          //Constructor - for initializing each NODE to zero.
                linklist(void)
                 { first=NULL;
                  } void
                insertnode(int);
                void delnode();
                void display();
};
                                        // Function to insert the new NODE
void linklist :: insertnode(int ele)
{
 NODE *newnode = new NODE;
 newnode->info = ele;
 newnode->link = first;
 first=newnode;
}
                                    // Function to delete the NODE
void linklist :: delnode()
{
 clrscr();
 NODE *temp=first;
 if(first==NULL) cout<<"THE
 LIST IS EMPTY!!!!!"; else
    { cout<<"THE ELEMENT DELETED IS:- "<<first->info;        first=first-
>link; delete temp; cout<<"\n\
       tDELETION SUCCESSFUL!!!!";
     }
  getch()
 ; }
```

```
void linklist :: display()
{
  clrscr();
  NODE *temp=first;
  if(first==NULL)
     cout<<"THE LIST IS
  EMPTY !!!!!! "; else
  { cout<<"THE ELEMENTS IN THE LINKEDLIST
      ARE:-";
      while(temp!=NULL)
     { cout<<temp-
       >info<<"      ";
       temp=temp->link;
      }
   }
  getch()
; }

void main()
{               int
    ele,choice;
    linklist obj;
    clrscr();
   do
   {
         clrscr();
         cout<<"MENU:-\n"; cout<<"\t(1)INSERTION \n\
         t(2)DELETION \n\t(3)DISPLAY\n\t(4)EXIT";
         cout<<"\nENTER YOUR CHOICE :";
         cin>>choice;

         switch(choice)
         {
            case 1:       clrscr();
                          cout<<"ENTER THE ELEMENT TO BE
                          INSERTED :"; cin>>ele;
                          obj.insertnode(ele); break;
```

```
            case 2:         obj.delnode();
            case 3:         break;
            case 4:

                            obj.display();
                            break;


                           cout<<"THANK YOU!!!!";
                            break;
          }
     } while(choice!
   =4);
   getch();
}
```

```
  MENU:(1)INSERTION (2)DELETION (3)DISPLAY (4)EXIT

ENTER YOUR CHOICE : 1 ENTER THE ELEMENT TO BE INSERTED : 2

ENTER YOUR CHOICE : 1 ENTER THE ELEMENT TO BE INSERTED : 20

ENTER YOUR CHOICE : 1 ENTER THE ELEMENT TO BE INSERTED : 40

ENTER YOUR CHOICE : 3 THE ELEMENTS IN THE LINKEDLIST ARE:- 40          20
2

ENTER YOUR CHOICE :2 THE ELEMENT DELETED IS:- 40
                        DELETION SUCCESSFUL!!!!

ENTER YOUR CHOICE :2 THE ELEMENT DELETED IS:- 20
                          DELETION SUCCESSFUL!!!!

ENTER YOUR CHOICE :2 THE ELEMENT DELETED IS:- 2
                          DELETION SUCCESSFUL!!!!

ENTER YOUR CHOICE :2
                          THE LIST IS EMPTY!!!!!

ENTER YOUR CHOICE :3
                          THE LIST IS EMPTY !!!!!!
  ENTER YOUR CHOICE :4
THANK YOU!!!! 5. Write a C++ program to create a template function for
quick sort and demonstrate the sorting of integers, doubles and
character datatypes.
```

```
#include<iostream.h>
#include<conio.h>

                    // Function template for exchanging(any type)
```

```
template <class
t> void swap(t
&x, t &y)
{ t
  temp;
  temp=
  x;
  x=y;
  y=tem
  p;
}

              // Function template for sorting(any type)
template <class t1> void
quick(t1 a[], int low, int
high)
{ t1
  key;
  int i,j,flag=0;
  if(low<high)
  { key=a[low];
           i=low+1;
           j=high;
           while(!flag)
           { while((a[i] <= key ) &&
                    (i<high)) i++;
                    while(a[j]>key)
                             j--;
                    if(i<j)
                         swap(a[i],a[j]
                         );
                    else
                          flag=
                          1;
             }
           swap(a[low],a[j])
           ;
           quick(a,low,high
           -1);
           quick(a,low+1,h
           igh);
     }
   retur
 n; }
```

```
void main()
{ int i,j,num,a[30],choice;
        double b[30];

       cout<<"MENU:-\n";
cout<<"\t1. INTEGERS\n\t2. DOUBLES\n\t3. Exit\
                                                    n";

        clrscr();
        cout<<"ENTER YOUR CHOICE ...:";
        cin>>choice;
         switch(choice)
         {
           case 1: cout<<"Enter how many elements are there in the
                    integer array=>"; cin>>num; cout<<"Enter the
                    Integer numbers       : ";
                    for(i=0;i<num;i++)
                         cin>>a[i];
                    quick(a,0,num-1) ;
                    cout<<"\n The Sorted List
                    is... : "; for(i=0;i<num;i+
                    +) cout<<a[i]<<"\t";
                    break;

          case 2: cout<<"Enter how many elements are there in double
                   vector array=> "; cin>>num; cout<<"Enter the
                   Floating point numbers : ";
                   for(i=0;i<num;i++)
                       cin>>b[i];
                   quick(b,0,num-1)            ;
                   cout<<"\nThe  Sorted  List
                   is...  :  ";  for(i=0;i<num;i+
                   +) cout<<b[i]<<"\t";
                   break;

         default : cout<<"THANK YOU!!!!";
                    break;
     }
   getch();
}
```

Run 1.

MENU:1.
INTEGERS
2.DOUBLES
3.Exit

ENTER YOUR CHOICE ... : 1
Enter how many elements are there in the integer
array=>5 Enter the Integer numbers : -3 5 -25 0 2

The Sorted List is...: -25-3 0 2 5
Run 2.

MENU:1.
INTEGERS
2.DOUBLES
3.Exit

ENTER YOUR CHOICE ... : 2
Enter how many elements are therein double vector array=> 4
Enter the Floating point numbers : 2.5 -3.6 -1.345 1.25

The Sorted List is... : -3.6-1.345 1.25 2.5

06. Write a C++ program to create a class called STACK using an array of integers. Implement the following operations by overloading the operators + & -.

i) s1 = s1 + element - Where s1 is a object of the class STACK &element is a integer to be pushed on top of the stack.
ii) s1 = s1-- Where s1 is a object of the class STACK, -- operator pops the element from the stack.

Handle the STACK empty and STACK full conditions. Also display the status and contents of the stack after each operation, by overloading the operator <<.

```
#include<iostream.h>
#include<conio.h>
class STACK
{ int a[100], size, top;
 public :
                    //Constructor with one argument to intialize the
                  stack to -1 //And the Max size of the stack to n that
                                given by the user.
        STACK(int n)
        { top=-1;
                size=
                n;
        }

        void display();
        friend STACK operator +
        (STACK , int); friend STACK
        operator --(STACK ); friend int
        overflow(STACK); friend int
        empty( STACK );
        friend ostream& operator <<(ostream& ,STACK );
};

STACK operator +(STACK s1, int ele)
{
   clrscr(); s1.a[+
   +s1.top]=ele;
   return (s1);
}

STACK operator --(STACK s1)
{
```

```
   clrscr();
        cout<<" \nTHE ELEMENT "<<s1.a[s1.top--] <<" HAS BEEN POPPED
OUT OF THE STACK.";
   getch();
   return(s1);
}

         //Functions to check the overflow &
underflow int overflow( STACK s1)
{ if(s1.top==(s1.siz
   e-1) ) return(1);
   else return(0);
}
int empty( STACK s1)
{ if(s1.top==-1)
  return(1);
 else
return(0); }

ostream& operator <<(ostream& print,STACK s1)
{ if(empty(s
  1))
   {
     clrscr();
     print<<"\n\tTHE STACK IS
     EMPTY!!!!!"; getch();
   }
   else
   {
    clrscr();
    print<<"\nSTACK STATUS :-";
    for(int i=s1.top; i>=0;i--) print<<"\
            t"<<s1.a[i];
    getch();
   }
 return(prin
t); }

void main()
{ int choice,element,n;
          clrscr();
                cout<<"\nEnter the size of the stack : "; cin>>n;
           STACK s1(n);
           do
```

```
        {
                clrscr();
                cout<<"MENU :-\n(1)PUSH\n(2)POP\n(3)DISPLAY\
                n(4)EXIT"; cout<<"\nENTER YOUR CHOICE: ";

                cin>>choice;
                switch(choice)
                { case 1:cout<<"\nENTER THE ELEMENT TO BE
                 PUSHED :";
                        int element;
                        cin>>element;
                        if(overflow(s1)) {
                                clrscr();
                                 cout<<"\n\t STACK
                        OVERFLOW!!!!! "; getch(); }
                        else
                                  s1=s1+elem
                                  ent;
                       cout<<s1;
                       break;

                case 2: if(empty(s1))
                        {
                               clrscr();
                                cout<<"\n THE STACK IS
                       EMPTY!!!!"; getch(); }
                       else s1=s1--;
                       cout<<s1;
                       break;

                case 3: cout<<s1;
                        break;

             case 4: cout<<"\nTHANKYOU!!!!!";
                         break;
               }
       } while(choice!
    =4);
 getch(); }Out
put:
Enter the size of the stack :      3

MENU :-
```

```
        (1)PUSH
        (2)POP
        (3)DISPLAY
        (4)EXIT

ENTER YOUR CHOICE: 1
ENTER THE ELEMENT TO BE PUSHED :2
STACK STATUS :- 2

ENTER YOUR CHOICE: 1
ENTER THE ELEMENT TO BE PUSHED :4
STACK STATUS :-4     2

ENTER YOUR CHOICE: 1
ENTER THE ELEMENT TO BE PUSHED :6
STACK STATUS :-6     4     2

ENTER YOUR CHOICE: 1
                STACK OVERFLOW!!!!! ;
STACK STATUS :- 6     4     2

ENTER YOUR CHOICE: 2
THE ELEMENT 6 HAS BEEN POPPED OUT OF THE STACK.
STACK STATUS :- 4     2

ENTER YOUR CHOICE: 2
THE ELEMENT 4 HAS BEEN POPPED OUT OF THE STACK.
STACK STATUS :- 2

ENTER YOUR CHOICE: 2
THE ELEMENT 2 HAS BEEN POPPED OUT OF THE
                STACK. THE STACK IS EMPTY!!!!!

ENTER YOUR CHOICE: 2
                THE STACK IS EMPTY!!!!1
STACK STATUS :- THE STACK IS
EMPTY!!!!!

ENTER YOUR CHOICE: 4
                 THANKYOU!!!!!
```

07. Write a C++ program, to create a class called as DATE. Accept two valid dates in the form of dd/mm/yyyy. Implement the following by

overloading the operators - & +. Display the results by overloading the operator << after each operation.

i) no_of_days = d1 - d2 where d1 & d2 are Date objects. Assume d1> = d2 andno_of_days is an integer.
ii) d1 = d1 + no_of_days where d1 is a DATE object and no-of-days is an integer.

```
#include <iostream.h>
#include <conio.h>

class date
 {
                int dd,mm,yy;

        public:
                void read()
                  {
                     cin>>dd>>mm>>yy;
                  } int operator -
                (date); date
                operator + (int);
 friend ostream &operator<<(ostream &put,date &d);
 };
 int
 b[13]={0,31,29,31,30,31,30,31,31,30,31,3
 0,31}; int
 a[13]={0,31,28,31,30,31,30,31,31,30,31,3
 0,31};

 int date::operator-(date d2)
        {      date      res;      int
          nod,noly=0,temp=d2.
          yy;

          for
          (temp;temp<yy;temp
          ++) if(temp%4==0)
          noly++;

          temp=yy; for
          (temp;temp>d2.yy;tem
          p--) if(temp%4==0)
          noly++;

          res.dd=dd-d2.dd;
```

```
          if(res.dd<0)
          { res.dd=dd+a[m
                m];
                res.mm=mm--;
          }
        res.mm=mm-d2.mm;      if(res.mm<0)
          { res.mm=mm+1
          2; res.yy=yy--; }
          res.yy=yy-d2.yy;
          if(yy<0) return -1;
          nod=res.dd+(res.yy*365);

          int months=d2.mm;
          for(int
          i=1;i<=res.mm;i++)
          nod+=a[months++];

          nod+=noly;
     return nod;
    }
date date::operator+(int ndays)
         { date
          d;
          d.dd=dd; d.mm=mm; d.yy=yy;
          for(int i=1;i<=ndays;i++)
                 {
                  d.dd++; if(d.yy
                  %4==0)
                   { if(d.dd>b[d.m
                        m])
                          { d.dd=1;
                           d.mm++;
                           }
                   }
                  else
                  { if(d.dd>a[d.m
                     m])
                          { d.dd=1;
                           d.mm++;
                          }
                  }
                  if(d.mm>1
                  2)
                        { d.mm=1;
```

```
                        d.yy++;
                       }
                }
         return d;
   } ostream &operator<<(ostream &print,date &d)
  {

        print<<d.dd<<"/"<<d.mm<<"/"<<d.yy;
        return
  print; }

void main()
  { date d1,d2;
        int ndays;
        clrscr();
        cout<<"Enter two dates, date1 should be greater
        than date2\n\n"; cout<<"Enter Date1 (dd mm
        yyyy)::"; d1.read(); cout<<"Enter Date2 (dd mm
        yyyy)::"; d2.read(); clrscr();
        ndays=d1-d2;
        cout<<"\n\nNumber of Days between the\n\n\t\tDate1 ::
        “<<d1<<"\nand"
            <<"\t\tDate2 :: " <<d2<<" is =>"<<ndays;
        cout<<"\n\n\nEnter no. of days to be added to
        the Date1 ::"; cin>>ndays; d1=d1+ndays;
        cout<<"\n\n\t\tDate1 after adding "<<ndays<<" days
        is ::"<<d1;

        getch();
  }
```

RUN 1.

ENTER THE FIRST DATE IN THE DD/MM/YYYY FORMAT : 1 1 2000 ENTER THE SECOND DATE IN THE SAME FORMAT : 1 1 1999

THE NO. OF DAYS IN BETWEEN THE TWO YEARS ARE : 365 ENTER THE NO. OF DAYS TO BE ADDED TO THE FIRST DATE : 0

THE DATE AFTER 0 DAYS IS : 1 / 1 / 2000

RUN 2.
-----ENTER THE FIRST DATE IN THE DD/MM/YYYY FORMAT : 1 1 1997 ENTER THE SECOND DATE IN THE SAME FORMAT : 1 1 1996

THE NO. OF DAYS IN BETWEEN THE TWO YEARS ARE : 366 ENTER THE NO. OF DAYS TO BE ADDED TO THE FIRST DATE : 5

THE DATE AFTER 5 DAYS IS : 6 / 1 / 1997

08. Write a C++ program, to create a class called MATRIX using two dimensional arrays of integers. Implement the following by overloading the operators = which checks the compatibility of two matrices to be added and subtracted. Perform the following by overloading + and - operators. Display the result by overloading the operator <<.

```
if ( m1 = m2 )
 {
    m3 = m1 + m2 ;
    m4 = m1 - m2 ;
 }

else
        Display 'error'.
```

Where m1,m2,m3,m4 are MATRIX objects.

```
#include<iostream.h>
#include<conio.h>
class MATRIX
{ int row,column,a[10]
        [10];

public :
        MATRIX(int r, int c)
        { row=r;
         column=
         c; }

        void getmatrix(); int
        operator ==(MATRIX);
        MATRIX operator +
        (MATRIX); MATRIX
        operator -(MATRIX);
        friend ostream& operator <<(ostream&, MATRIX &);
};

void MATRIX :: getmatrix()
{ for(int i=0;i<row;i+
  +) for(int
  j=0;j<column;j++)
   { cout<<"
   [ "<<i<<","<<j<<" ] :";
   cin>>a[i][j];
```

```
   } } int MATRIX :: operator
==(MATRIX m1)
{ if((m1.row == row) && (m1.column
  == column))
          return(1);
 else
return(0); }


MATRIX MATRIX :: operator +(MATRIX m1)
{
         MATRIX m2(row,column);
          for(int i=0 ; i<row ; i++)
           for(int j=0 ; j<column ; j++)
                 m2.a[i][j]=a[i][j]+m1.a[i][j];
           return m2;
}

MATRIX MATRIX :: operator -(MATRIX m1)
{
          MATRIX m2(row,column);
          for(int i=0 ; i<row ; i++)
           for(int j=0 ; j<column ; j++)
                 m2.a[i][j]=a[i][j]-m1.a[i][j];
          return
m2; }


ostream& operator <<( ostream& print, MATRIX &res)
{ for(int i=0;i<res.row;i+
         +)
          { for(int j=0;j<res.column;j+
                   +)
                    print<<res.a[i][j]<<"";
                    print<<endl ;
            }
         return
         print;
} void



main() {
```

```
 int m,n,p,q;
 clrscr();

 cout<<"Enter the order of the First matrix => ";
 cin>>m>>n;
 MATRIX
 m1(m,n);

 cout<<"Enter the order of the Second matrix=>
 "; cin>>p>>q;
 MATRIX m2(p,q);

 if(m1 == m2)
 { cout<<"\nEnter the elements of First matrix
         :- \n"; m1.getmatrix();

         cout<<"\nEnter the elements of Second matrix :-\n";
         m2.getmatrix();
         MATRIX m3(m,n), m4(m,n);

         m3=m1+m2; m4=m1-m2; cout<<m1<<"
  +      \n"<<m2<<"      =      \n"<<m3<<endl;
  cout<<m1<<" - \n"<<m2<<" = \n"<<m4; }

 else
  cout<<"\nTHE TWO MATRICES ARE NONCOMPATIBLE FOR
  ADDITION OR
              SUBTRACTION!!!!";
 getch();
}
```

Run 1.

Enter the order of the First matrix => 2 1
Enter the order of the Second matrix => 2 1

THE TWO MATRICES ARE NONCOMPATIBLE FOR ADDITION OR SUBTRACTION!!!!

Run 2.

```
------
Enter the order of the First matrix => 2 2
Enter the order of the Second matrix=> 2 2

Enter the elements of First
 matrix :[ 0,0 ] : 2
 [ 0,1 ] : 3
 [ 1,0 ] : 4
 [ 1,1 ] : 5

Enter the elements of Second
 matrix :[ 0,0 ] : 6
 [ 0,1 ] : 5
 [ 1,0 ] : 4
 [ 1,1 ] : 8

2   3
4   5
   +
6   5
4   8
   =
8   8
 8 13


2   3
4   5
   -
6   5
4   8
   =
-4 -2
 0 -3
```

09. Write a C++ program, to create a class called as OCTAL which has the characteristics of an octal number. Implement the following by writing an appropriate constructor and an overloaded operator +.

Display the OCTAL object by overloading the operator <<. Also display the values of k & y.

i) OCTAL h = x; Where h is OCTAL object and x is an integer. ii) int y = h + k; here h is an OCTAl and k is an integer.

```
#include<iostream.h>
#include<conio.h
>
#include<math.h
> class OCTAL
{ int oct[15] , count;
  public:
          OCTAL(int);
          int operator +(int);
          friend ostream& operator <<(ostream& ,OCTAL)
};

OCTAL :: OCTAL(int x)
{ int
 i=0,rem,a[15]
 ; while(x!=0)
  { rem=x%8;
  x=x/8; a[i+
  +]=rem; }
  count=i; int
  n=count-1;
  for(i=0;i<count
  ;i++)
  {
   oct[i]=a[n];
   n--;
  }
}

int OCTAL :: operator +(int k)
{ int x=0;int
   j=count-1; for(int
   i=0;i<count;i++)
   { x=x+oct[j]*pow(8,
         i);
         j--;
   }
   return(x+k);
```

```
}ostream& operator <<(ostream& print,OCTAL o)
{ for(int
 i=0;i<o.count;i++)
 print<<o.oct[i];
 return
print; }
void
main()

{
 clrscr();
 int x,k,y=0; cout<<"\nEnter the Integer value
 of x (In Decimal) : "; cin>>x;
 OCTAL h=OCTAL(x); cout<<"The corresponding Octal
 Equivalent of      "<< x <<"   is       : "<<h; cout<<"\n\
 nEnter the integer to be added to the Octal value : ";
 cin>>k; y=h+k; cout<<"\t\t\t"<<h<<" (Octal) + "<<k<<"
 (Decimal) = "<<y<<" (Decimal)"; getch();
}
```

Run 1.

```
------
Enter the Integer value of x (In Decimal)    : 10
The corresponding Octal Equivalent of 10  is       : 12

Enter the integer to be added to the Octal value : 10
                        12 (Octal) + 10 (Decimal) = 20
```

(Decimal) Run 2.

```
------
Enter the Integer value of x (In Decimal)    : 8
The corresponding Octal Equivalent of 8    is       : 10

Enter the integer to be added to the Octal value : 2
                        10 (Octal) + 2 (Decimal) = 10 (Decimal)
```

Run 3.

```
------
Enter the Integer value of x (In Decimal)    : 5
The corresponding Octal Equivalent of 5     is       : 5

Enter the integer to be added to the Octal value : 6
                        5 (Octal) + 6 (Decimal)= 11 (Decimal)
```

10. Write a C++ program, to create a class QUEUE, with add and delete member functions. Using it, implement a QUEUE of integers and doubles. Demonstrate the implementation by displaying the status and the content of the QUEUE after every operation.

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>

template <class
t> class QUEUE
{ t a[100]; int
        rear;
  public:
        QUEUE()
        { rear
        =0; }

        void add(t);
        void   del();
        void
        display();
};
template <class t> void
QUEUE <t> :: add( t ele)
{ a[+
  +rear]=ele
  ;
}

template <class t>
void QUEUE <t> ::
del()
{ if(rear=
 =0)
  { cout<<"QUEUE IS EMPTY !!!!!\n\n PRESS ANY KEY TO
  CONTINUE........."; getch(); }

  else
  { t ele=a[1]; for(int
       i=1;i<=rear;i++)
       a[i]=a[i+1];
        rear--;   cout<<" THE ELEMENT DELETED FROM THE QUEUE IS
```

```
:"<<ele;
        getch(
        );
  }
}

template <class t> void
QUEUE <t> :: display()
{
  clrscr();
  cout<<"THE CONTENTS OF THE QUEUE ARE :- ";
  if(rear==0)
  cout<<"The QUEUE is
  empty"; for(int i=1;i<=rear;i+
  +) cout<<"\t"<<a[i];
   getch()
; }

void main()
{
  QUEUE <int> iq1;
  QUEUE
  <double>dq1;
  double ele1; int
  ch,choice,ele;char
  c; clrscr();
  cout<<" Enter 1 to implent the QUEUE of doubles and \n"
        << "2 to implement the QUEUE of
  integers :   "; cin>>ch; do

  {
    clrscr();
    cout<<"\nMENU :- \n\n(1) ADD \n(2)DELETE\n(3)DISPLAY\n";
    cout<<"\nEnter your choice :";
    cin>>choice;

    switch(choice)
    { case 1:
      clrscr();
      if(ch==2)
              { cout<<"Enter the (intger type)element to
                   be added :"; cin>>ele; iq1.add(ele);
                   iq1.display();
.....
```

The using software is free version, you can upgrade it to the upgrade version.http://www.allimagetool.com

```
break; }
 else
  {

   cout<<"Enter the (double type)element to
   be added :"; cin>>ele1; dq1.add(ele1);
   dq1.display();
  }
 break;

case 2: if(ch==2)
        {
         iq1.del();
         iq1.display(
        ); }
        else
        {
          dq1.del();
          dq1.displa
        y(); } break;

case 3: if(ch==2)
         iq1.display();
        else
         dq1.display();
         break;

 default: exit(0);
  }
 }while(choice!
=3); getch(); }
```

Run1:

```
Enter 1 to implent the QUEUE of doubles and
2 to implement the QUEUE of integers       : 2

MENU :(1)
      ADD
      (2)DELETE
      (3)DISPLAY
Enter your choice :1
```

```
Enter the (intger type)element to be added :10
THE CONTENTS OF THE QUEUE ARE :-    10
Enter your choice :1
Enter the (intger type)element to be added :20
THE CONTENTS OF THE QUEUE ARE :-      10    20
Enter your choice :1
Enter the (intger type)element to be added :30
THE CONTENTS OF THE QUEUE ARE :-      10     20      30
Enter your choice :1
Enter the (intger type)element to be added :40
THE CONTENTS OF THE QUEUE ARE :-      10     20     30       40
Enter your choice :2
THE ELEMENT DELETED FROM THE QUEUE IS :10
THE CONTENTS OF THE QUEUE ARE :-     20    30      40
Enter your choice :2
THE ELEMENT DELETED FROM THE QUEUE IS :20
THE CONTENTS OF THE QUEUE ARE :-     30     40
Enter your choice :2
THE ELEMENT DELETED FROM THE QUEUE
IS :30 THE CONTENTS OF THE QUEUE ARE :-
40
Enter your choice :2
THE ELEMENT DELETED FROM THE QUEUE IS :40
THE CONTENTS OF THE QUEUE ARE :- The QUEUE
is empty

Run 2.
------
Enter 1 to implent the QUEUE of
doubles and 2 to implement the
QUEUE of integers :       1

MENU :(1)
        ADD
```

```
        (2)DELETE
        (3)DISPLAY
Enter your choice :1
Enter the (double type)element to be added :1.02
THE CONTENTS OF THE QUEUE ARE :-1.02
Enter your choice :1
Enter the (double type)element to be added :4.20
THE CONTENTS OF THE QUEUE ARE :-1.024.2

Enter your choice :2
THE ELEMENT DELETED FROM THE QUEUE IS :1.02
THE CONTENTS OF THE QUEUE ARE :-4.2
Enter your choice :3
THE CONTENTS OF THE QUEUE ARE :-4.2
```

11. Write a C++ program, to create a class DLIST (doubly Linked List) with member functions, insert at a specified position, and delete from a specified position. Demonstrate the implementation by displaying the status and content of the list after every operation.

```
#include<iostream.h>
#include<conio.h>

struct list
{ int data; struct
        list *llink;
        struct list
        *rlink;
};  typedef  struct

list   node;   class

DLIST

{ node *first;
        node
        *last; int
        count;

    public:
                DLIST()
                { first=last=NULL;
                count=0; }
```

```
                void insertnode(int ,int);
                void
                insertfirst(int);
                void shownode();
                void
                delnode(int);
};


void DLIST :: insertfirst(int data)
{ first=last=new node; first-
 >data=data; first->llink=first-
 >rlink=NULL; count++; }


void DLIST :: insertnode(int data,int position)
{
int i;
if(position>count+1)
    {               cout<<"\nINVALID
        POSITION!!!!!"; getch();
        return;
   } node
*newnode=new node;
newnode->data=data;
if(position==1)
{ newnode-
 >llink=NULL;
 newnode-
 >rlink=first; first-
 >llink=newnode;
 first=newnode;
 }
else if(position==count+1)
    { newnode-
        >rlink=NULL;
        newnode-
        >llink=last; last-
        >rlink=newnode;
        last=newnode;
    }
else
{
```

```
 int i; node
 *temp=first;
 for(i=1;i<=count
 ;i++)
 {
    if(i==position)
    { node *prev=temp-
        >llink; newnode-
        >llink=prev; prev-
        >rlink=newnode;
        newnode-
        >rlink=temp; temp-
        >llink=newnode;
     } temp=temp-
   >rlink;
  }
 } count++; cout<<data<<"HAS BEEN SUCCESFULLY INSERTED
IN POSITION : "<<position; getch(); }
void DLIST :: delnode(int position)
{
         int i;
          node
          *temp=first; int
          data;
          if(first==NULL)
          { cout<<"THE LIST IS ALREADY
           EMPTY!!!!!! "; getch();
           retur
         n; }

   else if(position>count)
          { cout<<"INVALID
                POSITION!!!!!!"; getch();
                return;
          }
  else if(position==1 && first==last)
          {
                data=first->data;
                delete first;
           first=last=NULL;
           }

  else if(position==count)
```

```
        { temp=last;
        last=temp->llink;
        last->rlink=NULL;
        data=temp->data;
        delete temp; }

   else if(position==1)
        { temp=first;
                first=temp-
                >rlink;
                data=temp-
                >data; first-
                >llink=NULL;
                delete temp;
        }
  else
  { for(i=1;i<=count;i+
        +)
        {
          if(position==i)

          { node *prev=temp-
                >llink; data=temp-
                >data; prev-
                >rlink=temp-
                >rlink; node
                *next=temp->rlink;
                next->llink=prev;
                delete temp;
           }
         }
        }
  cout<<"THE DATA "<<data<<"HAS BEEN SUCCESSFULLY
  DELETED FROM
      POSITION : "
  <<position<<"\n"; getch();
  count--; return;
}

void DLIST :: shownode()
{ node *temp=first;
 cout<<"\nLIST
 STATUS :-";
 if(temp==NULL)
```

```
{ cout<<"\nTHE LIST IS
EMPTY!!!!"; getch(); }
else
{
  cout<<"\nTHE NUMBER OF NODES IN THE LIST IS : "<<count
       <<"\nTHE CONTENTS OF THE LIST ARE :- ";
  while(temp!=NULL)
  { cout<<temp-
   >data<<"\t";
   temp=temp->rlink;
  }
  getch();
 }
}

void main()
{ DLIST list1;
 int
 flag=0,choice
 ; int
 position,data;
clrscr();
do
{ cout<<"\n\t\tMENU:-\n(1)INSERT NODE \n"
 <<"(2)DELETE NODE\n(3)DISPLAY STATUS\
 n(4)EXIT\n";
   cout<<"\nENTER YOUR CHOICE
  PLEASE(1-4) :"; cin>>choice; switch
  (choice)
  { case 1: cout<<"ENTER THE
    DATA :"; cin>>data;
            if(flag==0)
            {

             list1.insertfirst(data);
             flag=1;
            }
            else
            {
             cout<<"\nENTER THE POSITION WHERE THE DATA IS TO
             BE
                        INSERTED : ";
            cin>>position;
            list1.insertnode(data,position);
```

```
            }
            list1.shownod
            e(); break;

    case 2:cout<<"ENTER THE POSITION OF THE NODE TO BE
    DELETED : ";
            cin>>position;
            list1.delnode(posit
            ion);
            list1.shownode();
            break;

    case 3:list1.shownode();
            break;

    case 4: cout<<"\tTHANK YOU!!!!";
             break;
   }
 }
while(choice!
=4); getch(); }
```

Run1 :

```
        MENU:-
        (1)INSERT NODE
        (2)DELETE NODE
        (3)DISPLAY STATUS
        (4)EXIT

ENTER YOUR CHOICE PLEASE(1-4) :3

LIST STATUS :-                  THE LIST IS EMPTY

ENTER YOUR CHOICE PLEASE(1-4) :2
ENTER THE POSITION OF THE NODE TO BE DELETED : 1
                        THE LIST IS ALREADY EMPTY
LIST STATUS :-
                        THE LIST IS EMPTY

ENTER YOUR CHOICE PLEASE(1-4) :4
                        THANK YOU

ENTER YOUR CHOICE PLEASE(1-4) :1
```

```
ENTER THE DATA :5

LIST STATUS :-
THE NUMBER OF NODES IN THE LIST IS : 1
THE CONTENTS OF THE LIST ARE :-  5
ENTER YOUR CHOICE PLEASE(1-4) :1
ENTER THE DATA :2

ENTER THE POSITION WHERE THE DATA IS TO BE INSERTED : 1
2 HAS BEEN SUCCESFULLY INSERTED IN POSITION : 1
LIST STATUS :-
THE NUMBER OF NODES IN THE LIST IS : 2
THE CONTENTS OF THE LIST ARE :-    2    5
ENTER YOUR CHOICE PLEASE(1-4) :1
ENTER THE DATA :6

ENTER THE POSITION WHERE THE DATA IS TO BE INSERTED : 3
6  HAS BEEN SUCCESFULLY INSERTED IN POSITION : 3
        LIST STATUS :-
THE NUMBER OF NODES IN THE LIST IS : 3
THE CONTENTS OF THE LIST ARE :-    2    5    6

ENTER YOUR CHOICE PLEASE(1-4) :1
ENTER THE DATA :8

ENTER THE POSITION WHERE THE DATA IS TO BE INSERTED : 5

INVALID POSITION
        LIST STATUS :-
THE NUMBER OF NODES IN THE LIST IS : 3
THE CONTENTS OF THE LIST ARE :-    2    5    6
ENTER YOUR CHOICE PLEASE(1-4) :1
ENTER THE DATA :1

ENTER THE POSITION WHERE THE DATA IS TO BE INSERTED : 4
1 HAS BEEN SUCCESFULLY INSERTED IN POSITION : 4
        LIST STATUS :-
THE NUMBER OF NODES IN THE LIST IS : 4
THE CONTENTS OF THE LIST ARE :-    2    5    6    1
```

12. Write a C++ program to create a class called STUDENT with data members USN, Name & Age. Using inheritance, create the classes UGSTUDENT and PGSTUDENT having fields as Semester, Fees and Stipend. Enter the data for at least 5 students. Find the Semester wise average age for all UG and PG students separately.

```
#include<iostream.h>
#include<conio.h>
#include<string.h>

class student
```

```
{
  protected     char
  :             name[20];
                int  regno;
                int age;
  publi         void
 c: };          getstudent()
                ;
void student::getstudent()
{           cout<<"Enter Name of student :";
              cin>>name; cout<<"Enter
              Register Number :";
              cin>>regno;
            cout<<"Enter Age    :" ;
              cin>>age;
}

class ugstudent : public student
{ public:     int sem , fees ,
 stipend;
              void          getug();
friend void avarage(); };

void ugstudent :: getug()
{
              getstudent();
                cout<<"Enter
                Semester              :";
                cin>>sem;
                cout<<"Enter Fees :";
                cin>>fees;
                cout<<"Enter Stipend
                :" ; cin>>stipend;
}


class pgstudent : public student
{ public:     int sem , fees ,
 stipend;
              void getpg();
              friend void
              avarage();
};
```

```
void pgstudent :: getpg()
{               getstudent();
                  cout<<"Enter
                  Semester              :";
                  cin>>sem;
                  cout<<"Enter Fees :";
                  cin>>fees;
                  cout<<"Enter Stipend
                  :" ; cin>>stipend;
}

void avarage()
{       ugstudent ug[20]; int num,avg,i,j;
        cout<<"Enter the number of ug
        students:"; cin>>num; for
        (i=0;i<num;i++) ug[i].getug();
        cout<<"sem"<<" "<<"avarage"<<endl;

        for(i=1;i<=8;i++)
        { int sum=0,count=0;
                cout<<i<<"
                      ";
              for(j=0;j<num;j++)
                { if(ug[j].sem==i)
                          { count++;
                              sum=sum+ug[j]
                              .age;
                           }
                 }
                if(count==
                0)
                { cout<<"0"<<
                  endl;
                  continue;
                 }
               else
                  cout<<sum/
         count<<endl; }

        pgstudent pg[20]; cout<<"Enter
        the number of pg students:";
        cin>>num; for (i=0;i<num;i++)
            pg[i].getpg();
         cout<<"sem"<<" "<<"avarage"<<endl;
```

```
          for(i=1;i<=3;i++)
          { int
                sum=0,count=
                0; cout<<i<<"
                ";
               for(int j=0;j<num;j++)
                  { if(pg[j].sem=
                      =i)
                      { count++;
                         sum=sum +
                         pg[j].age;
                       }
                   }
                   if(count==
                   0)
                    { cout<<"0"<<e
                     ndl; continue; }
                    else
                             cout<<sum/count<
                             <endl;
           }
  }

void main()
{
         clrscr();
          avarage();
          getch();
}
```

Run 1:

```
          Enter the number of ug students:3
          Enter Name of student : Yogish
          Enter Register Number : 304167
        Enter Age         : 25
       Enter Semester    : 8
        Enter Fees       : 4000
       Enter Stipend     : 7600

          Enter Name of student : chenni
          Enter Register Number : 304157
        Enter Age         : 26
       Enter Semester    : 8
        Enter Fees       : 4000
```

```
Enter Stipend    : 7600

  Enter Name of student : rajesh
  Enter Register Number : 304178
 Enter Age         : 24
Enter Semester    : 3
 Enter Fees       : 3500
Enter Stipend     : 2500
  sem avarage
    1    0
    2    0
    3    24
    4    0
    5    0
    6    0
    7    0
    8    25

  Enter the number of pg students:3
  Enter Name of student : mandaara
  Enter Register Number : 3041
 Enter Age         : 28
Enter Semester    : 3
 Enter Fees       : 55000
Enter Stipend     : 9500

  Enter Name of student : Purohit
  Enter Register Number : 2031
 Enter Age         : 27
Enter Semester    : 1
  Enter Fees  : 55000
  Enter Stipend       : 5000

  Enter Name of student : Mohan
  Enter Register Number : 2037
 Enter Age         : 26

Enter Semester    : 1

 Enter Fees       : 55000

Enter Stipend     : 5000

sem avarage

1    26
```

2 0

3 28

12. Write a C++ program to create a class called STUDENT with data members USN, Name & Age. Using inheritance, create the classes UGSTUDENT and PGSTUDENT having fields as Semester, Fees and Stipend. Enter the data for at least 5 students. Find the Semester wise average age for all UG and PG students separately.

```
#include<iostream.h>
#include<conio.h
>
#include<stdio.h
> class student
{ public: char name[20];
        int regno,age; void
        readstudent();
};

void student::readstudent()
{ cout<<"Name : ";
cin>>name;
cout<<"Rregno : ";
cin>>regno;
cout<<"Age : ";
cin>>age; }

class ugstudent:public student
{ public :
                int
                semester,fees,stipend;
                void readugstudent();
};

class pgstudent:public student
{ public:
                int
                semester,fees,stipend;
                void readpgstudent();
};

void ugstudent::readugstudent()
```

```
{ readstudent();
cout<<"Semester : ";
cin>>semester;
cout<<"Fees : ";
cin>>fees;
cout<<"Stipend: ";
cin>>stipend; }

void pgstudent::readpgstudent()
{ readstudent();
        cout<<"Semest
        er : ";
        cin>>semester;
        cout<<"Fees : ";
        cin>>fees;
        cout<<"Stipend : ";
        cin>>stipend;
}

void main()
{ ugstudent ug[20];
        pgstudent
        pg[20]; int
        i,n;
        clrscr();
        cout<<"Enter the no of students\
        n"; cin>>n; cout<<"Enter the
        details of ug students\n";
        for(i=1;i<=n;i++)
        { cout<<"\n Enter the details of
                student:"<<i<<endl;
                ug[i].readugstudent();
        } cout<<"Enter the details of pg
                students\n";
                for(i=1;i<=n;i++)
                { cout<<"\n Enter the details of
                        student:"<<i<<endl;
                    pg[i].readpgstudent();
                } cout<<"\n The average ages of ug
        students semesterwise are\n"
             <<" semester \t average
        age"; for(int
        sem=1;sem<=8;sem++)
```

```
        { float sum=0; int
                found=0,count=0; for(int
                i=1;i<=n;i++)
                if(ug[i].semester==sem)
                        { sum=sum+ug
                        [i].age;
                        found=1;
                        count++;
                        }
                if(found==1)
                cout<<sem<<"\t
        "<<sum/count<<endl; }

        cout<<"\n The average ages of pg students semesterwise are\n"
              <<" semester \t average age";
        for(sem=1;sem<=4;sem++)
        { float sum=0; int
                found=0,count=0; for(int
                i=1;i<=n;i++)
                if(pg[i].semester==sem)
                        { sum=sum+pg
                        [i].age;
                        found=1;
                        count++;
                        }
                if(found==1)
   cout<<sem<<"\t" <<sum/count<<endl;
        }
         getch();
}
```

Run 1:

```
        Enter the number of students:3

        Enter the details of ug students
        Enter the details of student 1
        Name : Yogish
        Regno : 304167
       Age   : 25
      Semester  : 8
       Fees       : 4000
     Enter Stipend    : 7600

        Enter the details of student 2
```

Name : chenni
Regno : 304157
Age : 26
Semester : 8
Fees : 4000
Enter Stipend : 7600

Enter the details of student 3

Name : rajesh
Regno : 304178
Age : 24
Semester : 3
Fees : 3500
Stipend : 2500
Enter the details of pg students
Enter the details of student 1
Name: mandaara
Regno : 3041
Age : 28
Semester : 3
Fees : 55000
Stipend : 9500

Enter the details of student 2
Name : Purohit
Regno : 2031
Age : 27
Semester : 1
Fees : 55000
Stipend : 5000

Enter the details of student 3
Name : Mohan
Regno : 2037
Age : 26
Semester : 1
Fees : 55000 Stipend :
5000

The average ages of ug students
semesterwise are semester average age

3 24

```
   8              25
The average ages of pg students semesterwise
 are semester      average age

   1              26
   3              28
```

13. Write a C++ program to create a class called STRING and implement the following operations. Display the results after every operation by overloading the operator <<.
i) STRING s1= “VTU” ii) STRING s2=”BELGAUM” iii) STRING s3=s1+s2; (Use Copy Constructor).

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class STRING
{ char str[40];
  public:

        STRING();                   // Default Constructor
       STRING(char s[]);       // Constructor with parameter
         STRING(STRING &s); // Copy Constructor
         STRING operator +(STRING);
         friend ostream& operator <<(ostream&, STRING&);
};


STRING::STRING()
{
       str[0]='\0';
}


STRING::STRING( char s[])
{
        strcpy(str,s);
}


STRING::STRING(STRING &s)
{
```

```
        strcpy(str,s.str);
}

STRING STRING :: operator +(STRING s1)
{
         return(strcat(str,s1.str));
}

ostream& operator <<(ostream& print,STRING &s1)
{
 print<<s1.str;
 return
print; }


void main()
{
 clrscr();
 char name1[20],name2[20];
 cout<<"Enter two Names as
 your Wish :";
 cin>>name1>>name2; STRING
 s1(name1);
 //STRING s1="yogish"; cout<<"\
 nNAME1   :       "<<s1;

 STRING s2(name2);
 //STRING s2="mandya";
 cout<<"\nNAME2       :
       "<<s2; STRING
 s3=(s1+s2);
 cout<<"\n\nNAME3     :
       "<<s3; getch();
}
```

Output:

```
Enter two Names as your Wish :vtu
        belgaum

NAME1  :  vtu
NAME2  : belgaum
```

NAME3 : vtubelgaum

14. Write a C++ program to create a class called BIN_TREE with member functions inorder, preorder, postorder. Create a BIN_TREE object to demonstrate the traversals.

```
#include<iostream.h>
#include<conio.h>

struct dlist
{ int info; struct dlist
        *left; struct
        dlist *right;
}; typedef struct
dlist node;


class BIN_TREE
{ node *root;

  public:
        BIN_TREE()
        { root=NULL; }
        void gettree(); void
        preorder(node*,int);
        void
        postorder(node*,int)
        ; void
        inorder(node*,int);
};

void BIN_TREE :: gettree()
{  int  c;  node
  *newnode;
   node *prev=NULL;
   int i,n,ele;
   cout<<"ENTER THE NO. OF NODES IN THE TREE :";
```

```
  cin>>n; cout<<"\
  nEnter the element :";
  for(i=1;i<=n;i++)
  {
   newnode= new node;
   clrscr(); cin>>ele; newnode-
   >info=ele; newnode-
   >left=newnode->right=NULL;
   if(root==NULL)
   { root=newnode
   ; continue; }
   node
   *temp=root;
   while(temp!
   =NULL)
   { if(ele<temp-
        >info)
        { prev=te
          mp;
          temp=te
          mp-
          >left; }
        else
        { prev=temp;
          temp=temp-
          >right;
        }

   } if(ele<prev->info)
   prev->left=newnode;
   else prev-
            >right=newnode;
  }
}

void BIN_TREE :: preorder( node* temp,int c)
{
 if(c==0)
           temp=ro
           ot;
 c++; if(temp!
 =NULL)
 { cout<<"\t"<<temp-
           >info;
```

```
            preorder(temp-
            >left,c);
            preorder(temp->right,c);
  }
}

void BIN_TREE::postorder(node* temp,int c)
{
 if(c==0)
        temp=roo
        t;
 c++;

 if(temp!=NULL)
 { postorder(temp->left,c);
           postorder(temp-
           >right,c); cout<<"\
           t"<<temp->info;
   }
}

void BIN_TREE::inorder(node *temp,int c)
{
 if(c==0)
        temp=root;
 c++; if(temp!
 =NULL)
 { inorder(temp->left,c);
           cout<<"\t"<<temp-
           >info;
           inorder(temp->right,c);
  }
}


void main()
{
 BIN_TREE
 t1; int c=0;
 clrscr();
 t1.gettree();
 clrscr();
 cout<<"\n\nPREORDER DISPLAY :- ";
```

```
  t1.preorder(NULL,c); cout<<"\n\
  nPOSTORDER DISPLAY :- ";
  t1.postorder(NULL,c);
  cout<<"\n\nINORDER DISPLAY :- ";
  t1.inorder(NULL,c);
  getch();
}
```

Run 1.

ENTER THE NO. OF NODES IN THE TREE :
5 Enter the element : 1 2 3 5 4

PREORDER DISPLAY :- 1 2 3 5 4

POSTORDER DISPLAY :- 4 5 3 2 1

INORDER DISPLAY :- 1 2 3 4 5 Run 2.

ENTER THE NO. OF NODES IN THE TREE : 4

Enter the element :9 7 3 5

PREORDER DISPLAY :-9 7 3 5

POSTORDER DISPLAY :- 5 3 7 9

INORDER DISPLAY :- 3 5 7 9

15. Write a C++ program create a class EXPRESSION. Accept an arithmetic expression assumed to be in vaild INFIX form) and assign to EXPRESSION object. Convert the expression in the object to POSTFIX form by writing appropriate member functions. Display the results.

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
#include<ctype.h>

class EXPRESSION
{
        char infix[15],
        postfix[15]; int
        len,top,stack[15];
public:
        EXPRESSION( char s[15])
        { top=-1;
         strcpy(infix
         ,s);
        } void in_post();
         void push( char
         item)
         { stack[+
           +top]=item;
         } char
        pop()
         { return
 stack[top--]; } int
 priority( char item);
 void display(); };

int EXPRESSION :: priority( char item)
{
 if(item=='+'||
        item=='-') return
        2;
 if(item=='*'||
        item=='/')
        return 3;
 if(item=='$')
        return 4;
```

```
 if(item=='('||
item=='#') return
1; return 0; }


void EXPRESSION :: in_post()
{ int
 i=0,j=0;
 char
 temp;

 push('#');
 while(infix[i]!='\0')
 {
  if(isalpha(infix[i]))
  postfix[j++]=infix[i];
  else if(infix[i]=='(')
  push(infix[i]);
  else if(infix[i]==')')
  { temp=p
   op();
   while(temp!='(')
   { postfix[j+
        +]=temp;
        temp=pop();
   }
  }
  else
  { temp=pop();
   while(priority(temp)>=priority(
   infix[i]))
   { postfix[j++]=temp;
            temp=pop();
     }
     push(tem
     p);
     push(infix[i]);
   } i++; }
temp=pop();
while(temp!
='#')
{ postfix[j+
+]=temp;
temp=pop(); }
```

```
postfix[j++]='\0';
}



void EXPRESSION :: display()
{ cout<<"\n\nTHE ENTERED INFIX EXPRESSION IS :-\n"<<infix;
cout<<"\n\nTHE CORRESPONDING POSTFIX EXPRESSION IS :-\
n"<<postfix; getch(); }

void main()
{ char
  exp[15];
  clrscr();
  cout<<"ENTER THE VALID INFIX EXPRESSION :";
  cin>>exp;
  EXPRESSION exp1(exp);
  exp1.in_post();
  exp1.display();
}
```

Run 1.

ENTER THE VALID INFIX EXPRESSION :(a+b)*c

THE ENTERED INFIX EXPRESSION IS :- (a+b)*c

THE CORRESPONDING POSTFIX EXPRESSION

IS : ab+c*

Run 2.

ENTER THE VALID INFIX EXPRESSION :a+b*c

THE ENTERED INFIX EXPRESSION IS :-a+b*c

THE CORRESPONDING POSTFIX EXPRESSION IS :- abc*+

VIVA QUESTIONS

1. What are the differences between Procedural & Object Oriented Programming?
2. What are all the concepts to recognize a language as an OOPL?
3. Distinguish between Object & Class.
4. What is data abstraction, data hiding & data encapsulation?
5. List out the advantages of OOP.
6. What is inheritance & polymorphism?
7. What is function prototype? Why it is required is it necessary for C++.
8. Define the following functions.
   - Inline functions.
   - Friend function.
9. Differentiate between the following
   - Function overloading v/s Operator over loading
   - Inline function & a regular function.
   - Friend member function & member function of a class.
   - Class & a structure.
   - Macro & inline function.
   - Static & Dynamic binding.
10. Advantages & Disadvantages of inline function.
11. What is reference variable. Advantages of using reference variables?.
12. What are the key words that are present in c++ but not in C?.
13. What are the difference between C & C++?.
14. What are the rules we have to following while using default arguments?
15. What are the advantages of using default arguments?
16. What is friend function? Why it is required. Do friends violate encapsulation?
17. Define static & static member of a class.

18. Define New & Delete operator.
19. Define the following.
    - Constructor & Destructor when they are executed.
    - Copy constructor & Parameterized constructor.
20. List out the characteristics of a constructor.
21. Why inheritance is required?.
22. What are the diff. types of inheritance?
23. Difference between Public, Private & Protected access specifier.
24. Define Virtual Function. Why it is required?.
25. What is virtual base class? Why it is required?.
26. What are the changes we can notice for C to C++?
27. What is this pointer?
28. Is there any difference in typecasting compared to that of C.
29. Explain some ways that ambiguity can introduced when over loading functions.
30. What are default arguments? Why it is required.
31. How does friend operator function differ from member operator?
32. Why operator to be overloaded?
33. What are the different situations we can use const keyword.
34. What is scope resolution operator? Why it is required.
35. What are the different types of constructors?
36. How does the C++ compiler resolves the complicts in overloaded functions.
37. Explain the 3 steps of overloaded resolution.
38. Which are all the operators that we can't overload?
39. Define function template & class template.
40. What are the differences between function overloading, operator overloading & Function template.
41. What is template instantiation?

42. What is a pure virtual function?
43. Why name of our language is C++ why not ++C.
44. Difference between virtual base class with Virtual function.

45Differences between Static binding and dynamic binding.

……END……

Wish U Good Luck …